**Implementation No. 2**

Date: 31/08/2020

Name: **Tushar Nankani** Roll No: **1902112** Batch: **C23**

**AIM:** Control Structures and Console IO.

**THEORY:**

1. **Syntax of Control Structures:**

* Control structures are programming blocks that can change the path we take through those instructions.
* There are three kinds of control structures:
  + **Conditional Branches**, which we use for choosing between two or more paths. There are three types in Java: ***if/else/else if, ternary operator and switch.***
  + **Loops** that are used to iterate through multiple values/objects and repeatedly run specific code blocks. The basic loop types in Java are ***for, while and do while.***
  + **Branching Statements**, which are used to alter the flow of control in loops. There are two types in Java***: break and continue.***

**SYNTAX:**

* **IF ... ELSE statements:**

if (count > 2) {

System.out.println("Count is higher than 2");

} else {

System.out.println("Count is lower or equal than 2");

}

* **SWITCH case:**

int count = 3;

switch (count) {

case 0:

System.out.println("Count is equal to 0");

break;

case 1:

System.out.println("Count is equal to 1");

break;

default:

System.out.println("Count is either negative, or higher than 1");

break;

}

* **FOR, WHILE, DO..WHILE loops:**

for (int i = 1; i <= 50; i++) {

*// code to execute*;

}

int cnt = 1;

while (cnt <= 50) {

*// code to execute;*

cnt++;

}

do {

*// code to execute;*

} while (condition);

* **CONTINUE and BREAK**

int cnt;

for (cnt = 1; cnt < = 5; cnt++) {

if(cnt==2)

continue;

if (cnt==4)

break;

System.out.println("Current value of cnt is: " + cnt);

}

System.out.println("Current value of cnt is: " + cnt);

1. **BufferedReader and Scanner with Wrapper class:**

Scanner and BufferReader both classes are used to read input from external system. Scanner is normally used when we know input is of type string or of primitive types and BufferReader is used to read text from character streams while buffering the characters for efficient reading of characters.

***DECLARATION:***

* BufferedReader b = new BufferedReader(new InputStreamReader(System.in));
* Scanner s = new Scanner(System.in);

**KEY DIFFERENECES:**

| **Sr. No.** | **Key** | **Scanner Class** | **BufferedReader Class** |
| --- | --- | --- | --- |
| 1 | Synchronous | Scanner is not synchronous in nature and should be used only in single threaded case. | BufferReader is synchronous in nature. During multithreading environment, BufferReader should be used. |
| 2 | Buffer Memory | Scanner has little buffer of 1 KB char buffer. | BufferReader has large buffer of 8KB byte Buffer as compared to Scanner. |
| 3 | Processing Speed | Scanner is bit slower as it need to parse data as well. | BufferReader is faster than Scanner as it only reads a character stream. |
| 4 | Methods | Scanner has methods like nextInt(), nextShort() etc. | BufferReader has methods like parseInt(), parseShort() etc. |
| 5 | Read Line | Scanner has method nextLine() to read a line. | BufferReader has method readLine() to read a line. |

**PROGRAMS:**

1. **Write a program to print the grade as per the following criteria:**

|  |  |
| --- | --- |
| 0 – 60  61 – 70  71 – 80  81 – 90  91 – 100 | F  D  C  B  A |

**CODE:**

import java.io.\*;

class marks

{

      public static void main(String ar[])throws IOException

      {

        int marks;

        char grade = 'F';

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter the student's marks: ");

        marks=Integer.parseInt(br.readLine());

        if(marks > 90 && marks <= 100)

            grade = 'A';

        else if(marks > 80 && marks <= 90)

            grade = 'B';

        else if(marks > 70 && marks <= 80)

            grade = 'C';

        else if(marks > 60 && marks <= 70)

            grade = 'D';

        else if(marks >= 0 && marks <= 60)

            grade = 'F';

        else{

            System.out.println("INAVALID MARKS");

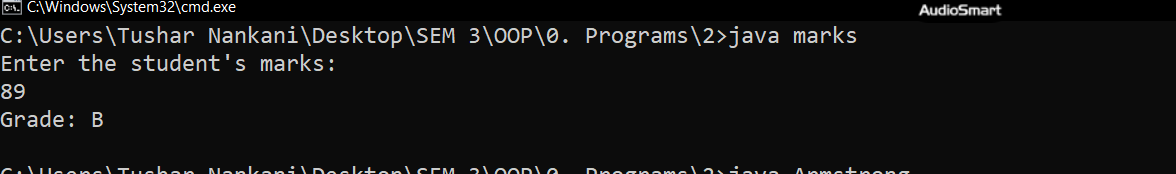
            System.exit(0);

        }

        System.out.println("\nGrade: " + grade);

      }

**OUTPUT:**

****

1. **Write a program to find Armstrong Numbers from 100 to n.**

**CODE:**

import java.io.\*;

import java.lang.\*;

class Armstrong

{

      public static void main(String ar[])throws IOException

      {

                int num, len, ans = 0, i, tmp;

                String numStr;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter a number: ");

        num = Integer.parseInt(br.readLine());

        for(i = 100; i <= num; i++)

        {

            len = 0;

            ans = 0;

            tmp = i;

            while(tmp != 0)

            {

                tmp /= 10;

                len += 1;

            }

            tmp = i;

            while(tmp != 0)

            {

                ans += Math.pow(tmp % 10, len);

                tmp /= 10;

            }

            if(i == ans)

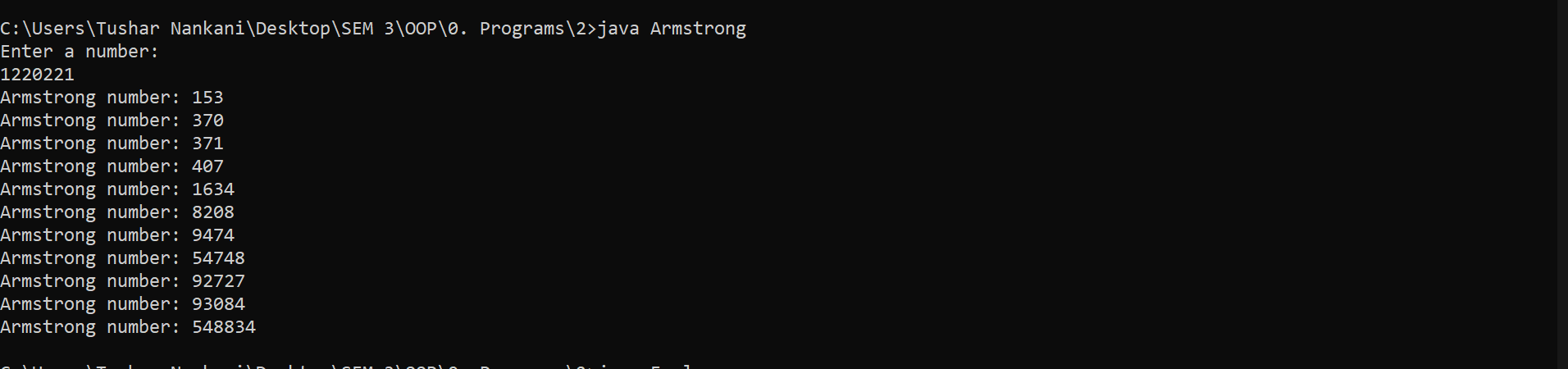
                System.out.println("Armstrong number: " + ans);

        }

      }

}

**OUTPUT:**

****

1. **Write a program to read and display details of an employee using single class and its object.**

**CODE:**

import java.io.\*;

import java.lang.\*;

class Employee

{

    String name, dept, des, id;

    void getEmp()throws IOException

    {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter a employee name: ");

        name = br.readLine();

        System.out.println("Enter a employee id: ");

        id = br.readLine();

        System.out.println("Enter a employee Department: ");

        dept = br.readLine();

        System.out.println("Enter a employee Designation:");

        des = br.readLine();

    }

    void putEmp()

    {

        System.out.println("Employee name: "+ name);

        System.out.println("Employee id: " + id);

        System.out.println("Employee Department: " + dept);

        System.out.println("Employee Designation: " + des);

    }

        public static void main(String ar[])throws IOException

        {

                Employee e = new Employee();

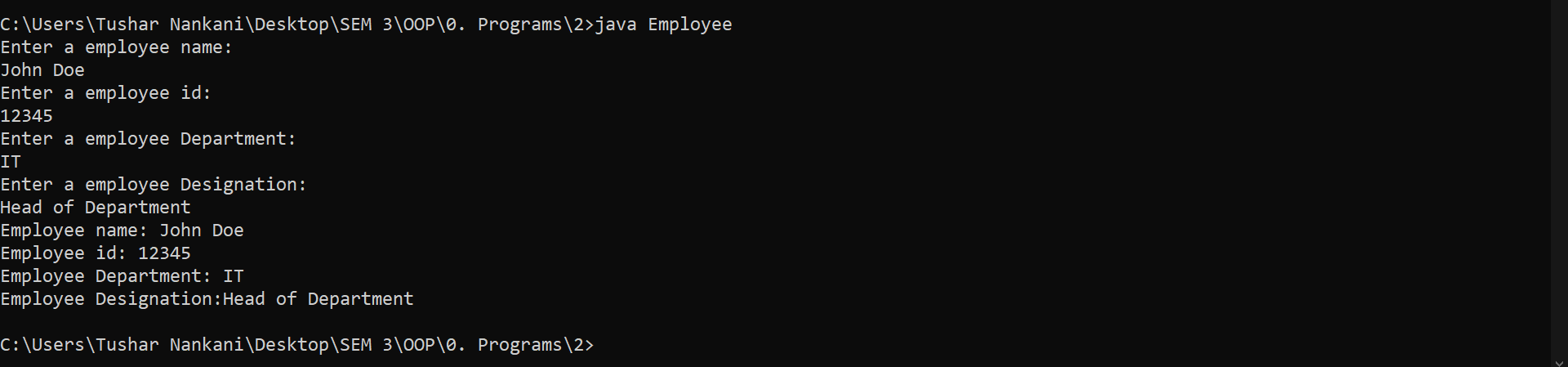
        e.getEmp();

        e.putEmp();

      }

}

**OUTPUT:**

****

1. **Write a program to find factorial of number using 2 classes and functions.**

**CODE:**

import java.io.\*;

import java.lang.\*;

class Factorial

{

    int f, num;

    void getNum()throws IOException

    {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter a number: ");

        num = Integer.parseInt(br.readLine());

    }

    void fact()

    {

        f = 1;

        for(int i = 2; i <= num; i++)

            f \*= i;

    }

    void printf()

    {

        System.out.println("The factorial of " + num + " is " + f);

    }

}

class MainFac{

    public static void main(String ar[])throws IOException

        {

                Factorial ans = new Factorial();

        ans.getNum();

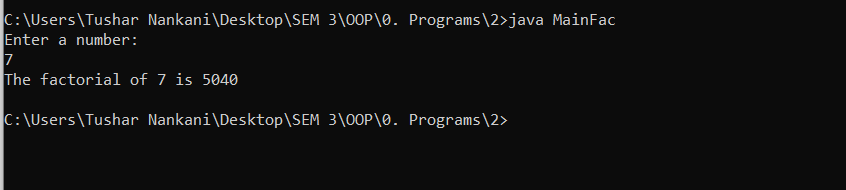
        ans.fact();

        ans.printf();

        }

}

**OUTPUT:**

****

1. **Write a program to find maximum of three numbers using conditional operator, using two classes and function returning result.**

**CODE:**

import java.io.\*;

import java.lang.\*;

import java.util.\*;

class Max3 {

    int x, y, z, max;

    Scanner s = new Scanner(System.in);

    void getNums()

    {

        System.out.print("Enter 3 numbers: \t");

        x = s.nextInt();

        y = s.nextInt();

        z = s.nextInt();

    }

    int getMax()

    {

        max = ((x > y) ? ((x > z) ? x : ((y > z) ? y : z)) :

((y > z) ? y : ((z > x) ? z : x)));

        return max;

    }

}

class Maximum

{

    public static void main(String args[])

    {

        Max3 m = new Max3();

        m.getNums();

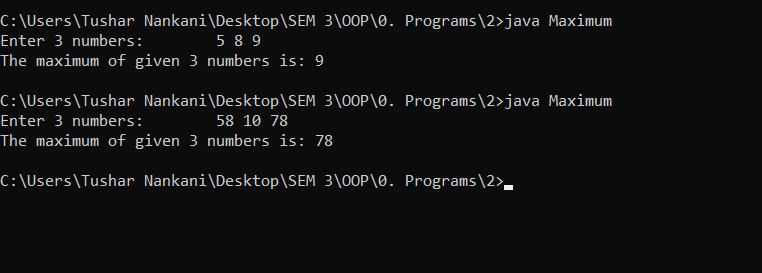
        int ans = m.getMax();

        System.out.println("The maximum of given 3 numbers is: "+ans);

    }

}

**OUTPUT:**

****